Curriculum Vitae

Dr. Klaus Havelund

Ph.D in Computer Science from the University of Copenhagen, Denmark.
Researcher at NASA Ames Research Center, California, USA.
Employed by Kestrel Technology.

Research Lab :
NASA Ames Research Center,
Moffett Field, CA 94035 USA
Email : havelund@ptolemy.arc.nasa.gov
Web : http://ase.arc.nasa.gov/havelund
Phone : (+1)650-604-3366.

Private Address :
2125 Rock Street, Apartment D
94043 Mountain View
California, USA.

Born the 17th of October 1955 in Denmark.

University Studies

1991–94 Ph.D in Computer Science from the University of Copenhagen,
Prepared at **Ecole Normale Supérieure, Paris.**
"The Fork Calculus - Towards A Logic for Concurrent ML".
Development of a process algebra in the CCS family with the objective to study a specification language for the Concurrent ML (CML) programming language (details p.2).

1986 Master Thesis in Computer Science from the University of Copenhagen.
"Stepwise Development of a Denotational Stack Semantics".
Study of the relationship between abstract and operational descriptions of programming languages (details p.3).

Professional Experience

Apr 1997 – present Researcher at NASA Ames Research Center, California, USA (details p.3).

Sep 1996–Mar 1997 Researcher at the Department of Computer Science, Aalborg University, Denmark (details p.3).

Oct 1994–Jul 1996 Researcher on a HCM (Human Capital Mobility) grant financed by the European Community. Research Lab: LITP, Paris 6, France (details p.4).

Jan–Oct 1994 Post-Doc at Ecole Polytechnique, Paris, France (details p.4).

1988–91 Researcher at CRI (advanced Danish software company) within the RAISE formal methods project, which in 1988 was transferred from DDC, see below (details p.4).

1984-88 Researcher at the industrial research institute DDC (Danish Datamatics Center) and member of the European ESPRIT project RAISE, the purpose of which was to develop a formal specification language (details p.4).

1979–82 Software programmer in various companies (half time during my university studies).

1 University Studies

1991–94 Ph.D in Computer Science from DIKU, Denmark

The Ph.D work was carried out from February 1991 to December 1993, and was defended March 1994 at DIKU, University of Copenhagen, Denmark. During this period I spent more than two years at Ecole Normale Supérieure, Paris, France.

The supervisor was Klaus Grue (DIKU). The members of the jury were Klaus Grue, Kim Guldstrand Larsen (AUC, Denmark) and Mogens Nielsen (DAIMI, Denmark).

The main motivation for writing a Ph.D was to extend the work I had carried out during 6 years in the European ESPRIT project RAISE.

The thesis, written in English, has as title: *"The Fork Calculus—Towards a Logic for Concurrent ML"*. The Fork Calculus, FC, presents a theory for dynamic process creation where processes interact through hand-shake communication. This calculus differs from Robin Milner's CCS in the way that processes are put in parallel. In CCS there exists a binary parallel operator `|` with which two processes `p` and `q` may be put in parallel as `p|q`. In FC this binary operator has been replaced with a unary `fork`-operator, and a process `p` can be activated to execute in the background, in parallel with the remaining program, with the command `fork(p)`. Also, FC has sequential composition instead of the action prefixing of CCS.

After having defined the syntax for FC, I have constructed an operational semantics, and based on that, I have studied various bisimulations, including a complete axiomatization of one of these. Two extensions of this calculus are then studied, one of which deals with program refinement, and one of which deals with dynamic process configuration as found in Robin Milner's π-calculus. For each of these three calculi I have defined a Hennessy-Milner like modal logic.

The three calculi shall be seen as approximations to defining a refinement logic for the programming language CML (Concurrent ML). CML is an extension of the programming language ML with concurrency primitives, amongst them a `fork`-operator. The thesis ends with an outline of such a logic for CML.

Part of the work has been carried out in collaboration with Kim Guldstrand Larsen, Aalborg University (AUC), Denmark. Hence, K. Larsen is co-author on published papers. The references from my thesis work are the following: [20, 21, 14]

1986 Master Thesis in Computer Science from DIKU, Denmark

The topic of the thesis [12] was denotational semantics of programming languages with professor Neil D. Jones as supervisor. I got the note : 9 out of 10 – corresponding to 11 on the Danish scale. The goal of the project was to bridge the gap between an abstract semantic definition of an Algol-like programming language, and a concrete operational definition of the same language. The bridge was created by a series of four intermediate semantics, getting more and more concrete. In particular the work revealed systematically the distinction between static (compile time) and dynamic (runtime) semantics.

During my studies I also wrote a syntax checker for the Meta-IV language, the specification language of the formal method VDM. This together with the above denotational semantics project was my real introduction to the area of formal methods that I have then stayed within for more than a decade now.

In general, the education at DIKU is planned as a five year study, with three years broad introduction to fundamental areas of computer science, followed by two years of more advanced topics, including the master thesis.

2 Professional Experience

Apr 1997– present Researcher at NASA Ames Research Center, California, USA

I currently work for Kestrel Technology at NASA Ames Research Center as a researcher. My activities concentrate on program verification and testing, and the goal is to develop techniques for locating errors in parallel programs. I work more specifically on runtime verification where program executions are monitored and checked against requirement specifications. I have performed one of the more successful applications of model checking: the analysis of the Remote Agent for the Deep-Space 1 space craft. I also regard myself as the idea generator and first prototype implementer of the Java PathFinder project, which is the major project in the Automated Software Engineering group today. Java PathFinder is a model checker for Java. This is pioneering work that has motivated other researchers on the international scene.

Sep 1996–Mar 1997 Researcher at Aalborg University, Denmark

I here worked in Kim Guldstrand Larsen's formal methods group. This group is known for their expertise in model checking, and in particular real-time model checking. During this stay, I applied the real-time model checker UPPAAL (developed partly by this group) to a 10 year old real-life audio/video protocol from the Audio/Video company Bang & Olufsen. During this effort I spotted the source of a known error, which had been around throughout all those years without being identifiable by normal testing.

Oct 1994–Jul 1996 HCM Grant, Paris 6 University, France

I was here financed by a HCM (Human Capital Mobility) grant from the European Community to do research in concurrency verification. My work in Therese Hardin's group was focused on formal specification and verification of concurrent real-time systems, for example communication protocols. Special emphasis was put on combining theorem proving in classical typed higher order logic with theorem proving in temporal logic. This was done basically by embedding TLA ('Temporal Logic of Actions' developed by Leslie Lamport) into the general purpose theorem prover PVS ('Prototype Verification System') developed by Owre, Shankar and Rushby at SRI International, California. A branch of this work consisted of combining theorem proving and model checking.

I collaborated with the people at SRI, in particular with Natarajan Shankar. A result of this collaboration is the paper [28] presented at the *Formal Methods Europe* conference at Oxford in March 1996. I have spent more than 5 months at SRI over the last 18 months.

As an additional result of this collaboration, I initiated a visit to Paris 6 by John Rushby, where he gave a one day PVS course for academics and industrial people. About 70 persons attended. My own contribution was to provide a "hands-on" practical exercise in using PVS on computers after the course of John Rushby. As a result of my stay at Paris 6, PVS is now used there in research (including a Ph.D student) and will soon be used in teaching. Also, the PVS system is now available via ftp from Paris 6, which then has become one of 3 European internet sites providing PVS in addition to SRI in California.

Jan–Oct 94 Post-Doc at Ecole Polytechnique, Paris

I worked as a post-doc in Radhia Cousot's group, financed by Ecole Polytechnique. During this period I learned about the theorem prover PVS, and considered how it could be used to specify and verify parallel systems.

1988–91 Researcher at CRI — the RAISE Project, Denmark

In 1988 the Danish software house CRI took over parts of the activities of DDC, amongst these the European ESPRIT project RAISE, see below. Hence, as participant of this project, and at that time employed by DDC, I was transferred from DDC to CRI. In general CRI was involved in several European ESPRIT projects, and in the European Space Agency's programs. Hence, an inspiring international environment.

1984–1988 Researcher at DDC — The RAISE Project, Denmark

DDC was an industrially oriented research institution, with main activities within ESPRIT projects (European research program for information technology). The initiator and scientific chief of the institute was Professor Dines Bjørner – DTU (Technical University of Denmark), currently director of the United Nations University for software technology in Macau.

I was working for 6 years as scientific staff in ESPRIT project 315: RAISE. RAISE stands for 'Rigorous Approach to Industrial Software Engineering'. The purpose of the project

was to produce a formally (mathematically) based method for producing software. The overall goal was to combine VDM ('Vienna Development Method' developed by Dines Bjørner and Cliff Jones) with CSP ('Communicating Sequential Processes' developed by Hoare). The major teams of the project were the Danish DDC team and the British STC (Standard Telephones and Cables) team. However, also ABB and ICL were involved in the project. The project covered approximately 100 man years, with the partition between Danish and British effort being around 50:50.

The project had a number of internationally known computer scientists associated as consultants, who continuously followed the project. These were Manfred Broy, Cliff Jones, Don Sannella and Andrzej Tarlecki.

My main responsibilities throughout the 6 years were language design and semantics of the resulting language. The language design was carried out by a group of 10 people, and was based on case studies generated by the true industrial partners (STC, ABB, ICL). After the language design, I wrote the final language semantics together with Robert Milne during a period of approximately 8 months. The last I did in the project was to write the majority (85%) of the textbook (published by Prentice-Hall) which explains the RAISE specification language. That is, I wrote *all* of the 250 page tutorial, and half of the 100 page reference manual. See [13]. This book is used today as a general introduction to formal methods at the Technical University of Denmark. Other RAISE references are [10, 6].

I have written several reports during the project, often in collaboration with other members of the team, and some of these were so-called deliverables to the European Commission.

The RAISE product (a specification language and associated tools) has been tested in the 5 year follow up ESPRIT project 5383: LaCoS, which involved several companies in several European countries: CRI, BNR Europe, SYPRO, Bull, MATRA Transport, INISEL Espacio, SSI, Technisystems and Lloyd's Register of Shiping.

See information about RAISE on the World Wide Web on the address: "http://dream.dai.ed.ac.uk/raise". I was not involved in LaCoS since I decided to do a Ph.D directly after the end of the RAISE project.

3 Editorial Board

On the editorial board for the International Journal of Software and Information Technologies (IJSIT) – `http://www.ijsit.org`.

4 Workshop and Conference Involvement

I have participated in the organization of workshops, as invited speaker and as a PC member for various workshops and conferences as stated below.

- Organized (on invitation) the 7th International SPIN workshop in year 2000 (SPIN'00) at Stanford University, California. The workshop lasted 3 days and was stand-alone (was not associated to a bigger conference).

- Took initiative and organized a sequence of workshops on Runtime Verification. RV'01 was held as a CAV'01 satellite event in Paris, France, July 2001. RV'02 was held as a CAV'02 satellite event in Copenhagen, Denmark, July 2002. RV'03 was held as a CAV'03 satellite event in Colorado, USA, July 2003. RV'04 was held as an ETAPS'04 satellite event in Barcelona, Spain, April 2004.
- Invited speaker at ASM 2003, International Workshop on Abstract State Machines.
- Program committee member on the following workshops and conferences:
  - ICI 2004, International Conference on Informatics.
  - CAV 2004, The 16th Conference in Computer Aided Verification.
  - DAW 2004, Dynamic Aspects Workshop.
  - SFEDL 2004, Semantic Foundations of Engineering Design Languages.
  - REOS 2003, Workshop on Requirements Engineering and Open Systems.
  - ASARTI 2003, Workshop: Advancing the State-of-the-Art in Run-Time Inspection.
  - FMPPTA 2003, The 8th International Workshop on Formal Methods for Parallel Programming: Theory and Applications.
  - DSN'03, Workshop on Model Checking for Dependable Software-Intensive Systems.
  - ESEC/FSE'03, 4th joint meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering.
  - CAV 2002, The 14th Conference in Computer Aided Verification.
  - RT-TOOLS, 2002 Workshop on Real-Time Tools.
  - FMPPTA 2002, The 7th International Workshop on Formal Methods for Parallel Programming: Theory and Applications.
  - SPIN 2002, The 9th International SPIN Workshop on Model Checking of Software.
  - SPIN 2001, The 8th International SPIN Workshop on Model Checking of Software.
  - FMPPTA 2001, 6th International Workshop on Formal Methods for Parallel Programming: Theory and Applications.
  - MVI 2001, Model-based Validation of Intelligence.
  - Lfm 2000, Fifth NASA Langley Formal Methods Workshop.
  - PATV 2000, The First International Workshop on Automated Program Analysis, Testing and Verification.
  - JFLA 2000, Les onzimes Journes Francophones des Langages Applicatifs.

5 Awards

Won EASST award for best software science paper presented at ETAPS'02. Paper title: "Synthesizing Monitors for Safety Properties", co-authored with G. Roşu [27].

References

[1] C. Artho, A. Biere, and K. Havelund. High-Level Data Races. In *VVEIS'03, The First International Workshop on Verification and Validation of Enterprise Information Systems*, April 2003. Angers, France.

[2] C. Artho, D. Drusinsky, A. Goldberg, K. Havelund, M. Lowry, C. Pasareanu, G. Roşu, and W. Visser. Experiments with Test Case Generation and Runtime Analysis. In E. Börger, A. Gargantini, and E. Riccobene, editors, *Abstract State Machines (ASM'03)*, volume 2589 of *LNCS*, pages 87–107. Springer, March 2003.

[3] C. Artho and K. Havelund. Applying Jlint to Space Exploration Software. In B. Steffen and G. Levi, editors, *Proceedings of Fifth International VMCAI conference (VMCAI'04)*, volume 2937 of *LNCS*. Springer, January 2004.

[4] H. Barringer, A. Goldberg, K. Havelund, and K. Sen. Program Monitoring with LTL in Eagle. In *Proceedings of Workshop on Parallel and Distributed Systems: Testing and Debugging*, April 2004.

[5] H. Barringer, A. Goldberg, K. Havelund, and K. Sen. Rule-Based Runtime Verification. In B. Steffen and G. Levi, editors, *Proceedings of Fifth International VMCAI conference (VMCAI'04)*, volume 2937 of *LNCS*. Springer, January 2004.

[6] D. Bjørner, A. Haxthausen, and K. Havelund. Formal, Model-oriented Software Development Methods: From VDM to ProCoS and from RAISE to LaCoS. *Future Generation Computer Systems*, 7, 1992.

[7] W. Buntine, B. Fischer, K. Havelund, M. Lowry, T. Pressburger, S. Roach, P. Robinson, and J. V. Baalen. Transformation Systems at NASA Ames. In *Software Transformation Systems*, May 1999. Los Angeles, California, USA.

[8] R. E. Filman and K. Havelund. Realizing Aspects by Transforming for Events. In *Proceedings, International Conference on Automated Software Engineering (ASE'02)*. Institute of Electrical and Electronics Engineers, 2002. Edinburgh, Scottland, 23-27 September 2002.

[9] R. E. Filman and K. Havelund. Source-Code Instrumentation and Quantification of Events. In Gary T. Leavens and Ron Cytron, editors, *Foundations Of Aspect-Oriented Languages, FOAL 2002, Enschede, The Netherlands*, April 2002. Proceedings available at `http://www.cs.wustl.edu/~cytron/FOAL`.

[10] C. George, K. Havelund, M. Nielsen, and K. Wagner. The RAISE Language, Method and Tools. *Formal Aspects of Computing*, 1(1), January-March 1989. Also published in Lecture Notes in Computer Science 328, Springer-Verlag, 1988.

[11] D. Giannakopoulou and K. Havelund. Automata-Based Verification of Temporal Properties on Running Programs. In *Proceedings, International Conference on Automated Software Engineering (ASE'01)*, pages 412–416. Institute of Electrical and Electronics Engineers, 2001. Coronado Island, California, 26-29 November 2001.

[12] K. Havelund. Stepwise Development of a Denotational Stack Semantics. Technical report, DIKU, Department of Computer Science, University of Copenhagen, Denmark, 1984.

[13] K. Havelund. *RSL Tutorial.* The BCS Practitioner Series. Prentice Hall, 1992. The written material is part I of the book named 'The RAISE Specification Language'. Part II of the book (pages 251–369) is the 'RSL Reference Description' and is written by A. Haxthausen and K. Havelund in collaboration.

[14] K. Havelund. *The Fork Calculus – Towards a Logic for Concurrent ML.* PhD thesis, Institute for Computer Science – University of Copenhagen (DIKU), March 1994. DIKU technical report 94/4.

[15] K. Havelund. Java PathFinder, A Translator from Java to Promela. In *Theoretical and Practical Aspects of SPIN Model Checking – 5th and 6th International SPIN Workshops*, number 1680 in Lecture Notes in Computer Science. Springer-Verlag, July and September 1999. Trento, Italy – Toulouse, France (presented at the 6th Workshop).

[16] K. Havelund. Java PathFinder User Guide. Technical Report, NASA Ames. Documents the implemented tool, 1999.

[17] K. Havelund. Mechanical Verification of a Garbage Collector. In D. Méry and B. Sanders, editors, *FMPPTA'99: Fourth International Workshop on Formal Methods for Parallel Programming : Theory and Applications*, number 1586 in Lecture Notes in Computer Science, pages 1258–1283. Springer-Verlag, April 1999. San Juan, Puerto Rico, USA.

[18] K. Havelund, A. Goldberg, R. E. Filman, and G. Roşu. Program Instrumentation and Trace Analysis. In *Monterey Workshop 2002, Radical Innovations of Software and Systems Engineering in the Future, Venice, Italy*, October 2002. Invited participation. One page abstract, full version to appear in Lecture Notes in Computer Science.

[19] K. Havelund, S. Johnson, and G. Roşu. Specification and Error Pattern Based Program Monitoring. In *European Space Agency Workshop on On-Board Autonomy, Noordwijk, Holland.*, pages 323–330. European Space Agency, October 2001.

[20] K. Havelund and K. G. Larsen. The Fork Calculus. *Nordic Journal of Computing*, 1, 1994. Also published in 20th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Computer Science 700, Springer-Verlag, 1993 and in a preliminary version in proceedings of 4th Nordic Workshop on Program Correctness, Report no 78, University of Bergen, 1993.

[21] K. Havelund and K. G. Larsen. A Refinement Logic for the Fork Calculus. In S. T. Vuong and S. T. Chanson, editors, *Protocol Specification, Testing and Verification XIV*, IFIP WG 6.1 Symposium. Chapman and Hall, 1995. Vancouver, Canada.

[22] K. Havelund and K. G. Larsen. StateText - A Textual Language for State Charts with Data. Technical Report, NASA Ames. To be published in revised version, 1997.

[23] K. Havelund, K. G. Larsen, and A. Skou. Formal Verification of an Audio/Video Power Controller using the Real-Time Model Checker UPPAAL. In *5th Int. AMAST Workshop on Real-Time and Probabilistic Systems*, number 1601 in Lecture Notes in Computer Science. Springer-Verlag, May 1999. Bamberg, Germany.

[24] K. Havelund, M. Lowry, S. Park, C. Pecheur, Penix J, W. Visser, and J. L. White. Formal Analysis of the Remote Agent, Before and After Flight. In *Fifth NASA Langley Formal Methods Workshop, Virginia*, June 2000.

[25] K. Havelund, M. Lowry, and J. Penix. Formal analysis of a space craft controller using SPIN. *IEEE Transactions on Software Engineering*, 27(8):749–765, August 2001. First version appeared in: Proceedings of the 4th SPIN workshop, Paris, France, November 1998.

[26] K. Havelund and T. Pressburger. Model Checking Java Programs using Java PathFinder. *International Journal on Software Tools for Technology Transfer*, 2(4):366–381, April 2000. Special issue of STTT containing selected submissions to the 4th SPIN workshop, Paris, France, 1998.

[27] K. Havelund and G. Roşu. Synthesizing Monitors for Safety Properties. In Joost-Pieter Katoen and Perdita Stevens, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2002, Grenoble, France*, Electronic Notes in Theoretical Computer Science, pages 342–356. Springer, April 2002. **Won EASST award for best software science paper presented at ETAPS'02**.

[28] K. Havelund and N. Shankar. Experiments in Theorem Proving and Model Checking for Protocol Verification. In M-C. Gaudel and J. Woodcock, editors, *FME'96: Industrial Benefit and Advances in Formal Methods*, volume 1051 of *Lecture Notes in Computer Science*, pages 662–681. Springer-Verlag, 1996.

[29] K. Havelund and N. Shankar. A Mechanized Refinement Proof for a Garbage Collector. NASA Ames Research Center. To be published, 1998.

[30] K. Havelund and J. Skakkebæk. Applying model checking in Java verification. In *Theoretical and Practical Aspects of SPIN Model Checking – 5th and 6th International SPIN Workshops*, number 1680 in Lecture Notes in Computer Science. Springer-Verlag, July and September 1999. Trento, Italy – Toulouse, France (presented at the 6th Workshop).

[31] K. Havelund, A. Skou, K. G. Larsen, and K. Lund. Formal Modeling and Analysis of an Audio/Video Protocol: An Industrial Case Study Using UPPAAL. In *Proc. of the 18*th *IEEE Real-Time Systems Symposium*, pages 2–13, Dec 1997. San Francisco, California, USA.

[32] K. Havelund, S. Stoller, and S. Ur. Benchmark and Framework for Encouraging Research on Multi-Threaded Testing Tools. In *PADTAD'03, Parallel and Distributed Systems: Testing and Debugging*, April 2003. Nice, France. Invited paper.

[33] K. Havelund and W. Visser. Program Model Checking as a New Trend. *International Journal on Software Tools for Technology Transfer*, 4(1), October 2002. Special issue of STTT containing selected submissions to the 7th SPIN workshop, Stanford, USA, 2000, organzied by K. Havelund, J. Penix and W. Visser. The article is the introductory article to the special issue. Havelund and Visser are guest editors.

[34] Klaus Havelund. Using Runtime Analysis to Guide Model Checking of Java Programs. In Klaus Havelund, John Penix, and Willem Visser, editors, *SPIN Model Checking*

and Software Verification, volume 1885 of *Lecture Notes in Computer Science*, pages 245–264. Springer, 2000.

[35] Klaus Havelund and Grigore Roşu. Java pathexplorer - a runtime verification tool. In *The 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (ISAIRAS'01)*, 2001. Montreal, Canada.

[36] Klaus Havelund and Grigore Roşu. Monitoring java programs with java pathexplorer. In Klaus Havelund and Grigore Roşu, editors, *Proceedings of the First Workshop on Runtime Verification (RV'01), Paris, France, July 2001*, volume 55 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science, 2001.

[37] Klaus Havelund and Grigore Roşu. Monitoring programs using rewriting. In *Proceedings, International Conference on Automated Software Engineering (ASE'01)*, pages 135–143. Institute of Electrical and Electronics Engineers, 2001. Coronado Island, California, 26-29 November 2001.

[38] Klaus Havelund and Grigore Roşu. Testing linear temporal logic formulae on finite execution traces. Technical Report TR 01-08, RIACS, May 2001. Written 20 December 2000.

[39] C. Pecheur J. Penix and K. Havelund. Using Model Checking to Validate AI Planner Domain Models. In *Proceedings of the 23rd Annual Software Engineering Workshop*, December 1998. NASA Goddard.

[40] L. Khatib, N. Muscettola, and K. Havelund. Verification of Plan Models using UPPAAL. In *First Goddard Workshop on Formal Approaches to Agent-Based Systems*, March 2000. NASA's Goddard Space center, Maryland, March 2000.

[41] M. Lowry, K. Havelund, and J. Penix. Verification and Validation of AI Systems that Control Deep-Space Spacecraft. In Z.W. Ras and A. Skowron, editors, *Tenth International Symposium on Methodologies for Intelligent Systems*, number 1325 in Lecture Notes in Artificial Intelligence. Springer-Verlag, October 1997. Charlotte, North Carolina.

[42] J. Penix, P. Alexander, and K. Havelund. Declarative Specification of Software Architectures. In *Proc. 12th IEEE International Conference on Automated Software Engineering*, November 1997. Incline Village.

[43] Grigore Roşu and Klaus Havelund. Synthesizing dynamic programming algorithms from linear temporal logic formulae. Technical Report TR 01-15, RIACS, May 2001. Written January 2001.

[44] W. Visser, K. Havelund, G. Brat, and S. Park. Java PathFinder - Second Generation of a Java Model Checker. In *Workshop on Advances in Verification (WAVE'00), Chicago, USA*. July 2000. Workshop associated with CAV'00.

[45] W. Visser, K. Havelund, G. Brat, and S. Park. Model Checking Programs. In *Proceedings of ASE'2000: The 15th IEEE International Conference on Automated Software Engineering*. IEEE CS Press, September 2000.

[46] W. Visser, K. Havelund, G. Brat, and S. Park. Model Checking Programs. *International Journal on Automated Software Engineering*, 2002. Extended version of paper presented at the conference on Automated Software Engineering 2000 (ASE'00).

[47] W. Visser, K. Havelund, and J. Penix. Adding Active Objects to SPIN. In *Theoretical and Practical Aspects of SPIN Model Checking – 5th and 6th International SPIN Workshops*, number 1680 in Lecture Notes in Computer Science. Springer-Verlag, July and September 1999. Trento, Italy – Toulouse, France (presented at the 5th Workshop).